Graphs that are simultaneously efficient open domination and efficient closed domination graphs

Sandi Klavžar[a,b,c], Iztok Peterin[c,d], Ismael G. Yero[e]

[a]*Faculty of Mathematics and Physics, University of Ljubljana, Slovenia*
[b]*Faculty of Natural Sciences and Mathematics, University of Maribor, Slovenia*
[c]*Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia*
[d]*Faculty of Electrical Engineering and Computer Science, University of Maribor, Slovenia*
[e]*Departamento de Matemáticas, EPS Algeciras, Universidad de Cádiz, Spain*

Abstract

A graph is an efficient open (resp. closed) domination graph if there exists a subset of vertices whose open (resp. closed) neighborhoods partition its vertex set. Graphs that are efficient open as well as efficient closed (shortly EOCD graphs) are investigated. The structure of EOCD graphs with respect to their efficient open and efficient closed dominating sets is explained. It is shown that the decision problem regarding whether a graph is an EOCD graph is an NP-complete problem. A recursive description that constructs all EOCD trees is given and EOCD graphs are characterized among the Sierpiński graphs.

Keywords: efficient open domination, efficient closed domination, perfect code, computational complexity, tree, Sierpiński graph
2010 MSC: 05C69, 05C05, 68R10, 94B60

Email addresses: `sandi.klavzar@fmf.uni-lj.si` (Sandi Klavžar), `iztok.peterin@um.si` (Iztok Peterin), `ismael.gonzalez@uca.es` (Ismael G. Yero)

[1]S.K. and I.P. acknowledge the financial support from the Slovenian Research Agency (research code funding No. J1-7110 and P1-0297).

[2]Most of the research was done while I.G.Y. was visiting the University of Maribor, Slovenia, as part of the grant "Internationalisation—a pillar of development of University of Maribor".

1. Introduction

The *domination number*, $\gamma(G)$, of a graph G is an important classical graph invariant with many applications. It is defined as the minimum cardinality of a subset of vertices S, called *dominating set*, with the property that each vertex from $V(G) - S$ has a neighbor in S. A dominating set S of cardinality $\gamma(G)$ is called a $\gamma(G)$-*set*. The union of closed neighborhoods centered at vertices of a dominating set covers the entire vertex set. A classical question for a cover of a set is: when does this cover form a partition? A graph G is called an *efficient closed domination graph*, or *ECD graph* for short, if there exists a set P, $P \subseteq V(G)$, such that the closed neighborhoods centered at vertices of P partition $V(G)$. Such a set P is called a *perfect code* of G. More general, a set P is an r-*perfect code* of G if the r-balls centered at vertices of P partition $V(G)$.

The study of perfect codes in graphs was initiated by Biggs [5] and presents a generalization of the problem of the existence of (classical) error-correcting codes. The initial research focused on distance regular and related classes of graphs, while later the investigation was extended to general graphs, cf. [33]. To determine whether a given graph has a 1-perfect code is an NP-complete problem [3] and remains NP-complete on k-regular graphs ($k \geq 4$) [34], on planar graphs of maximum degree 3 [13, 34], as well as on bipartite and chordal graphs [40]. On the positive side, the existence of a 1-perfect code can be decided in polynomial time on trees [13], interval graphs [35], and circular-arc graphs [29].

Recently the study of perfect codes in graphs was primarily focused on their existence and construction in some central families of graphs. Much research was done on standard graph products and product-like graphs [2, 23, 28, 39, 42, 44]. Among other classes of graphs on which perfect codes were investigated we mention Sierpiński graphs [8, 27], cubic vertex-transitive graphs [31], circulant graphs [10], twisted tori [24], dual cubes [25], and AT-free and dually chordal graphs [4].

A graph invariant closely related to the domination number is the *total domination number* $\gamma_t(G)$ [21]. It is defined as the minimum cardinality of a subset of vertices D, called a *total dominating set*, such that each vertex from $V(G)$ has a neighbor in D. A total dominating set D of cardinality $\gamma_t(G)$ is called a $\gamma_t(G)$-*set*. If we switch to neighborhoods, the union of open neighborhoods centered at vertices of a total dominating set covers the entire vertex set and again one can pose the question: when does this cover form a partition? A graph G is called an *efficient open domination graph*, or an *EOD graph* for short, if there exists a set D, $D \subseteq V(G)$, such that open neighborhoods centered at vertices of D partition $V(G)$. Such a set D is called an *EOD set*. Note that two different vertices of an EOD set are either adjacent or at distance at least 3.

The problem of deciding whether a graph G is an EOD graph is NP-complete [16, 38]. For various properties of EOD graphs see [15], a recursive characterization of EOD trees is given in [16]. EOD graphs that are also Cayley graphs were studied in [41], while EOD grid graphs were investigated in [7, 9, 30]. EOD direct product graphs were characterized in [1], for other standard graph products (lexicographic, strong, disjunctive and Cartesian) see [36]. Domination-type problems studied on graph products are usually most difficult on the Cartesian product, recall the famous Vizing's conjecture [6]. It is hence not surprising that EOD graphs studied on product graphs seems to be the most difficult on the Cartesian product. For some very recent results in this direction see [32].

In this paper we study the graphs that are ECD and EOD at the same time and call them *efficient open closed domination graphs*, *EOCD graphs* for short. In the rest of the paper we shall use the term *ECD set* instead of 1-perfect code to make the notation consistent.

We proceed as follows. In the rest of this section additional definitions are given and a basic result recalled. In the next section we show how to construct an ECD graph from and EOD graph and vice versa, and consider the structure of EOCD graphs from the viewpoint of the relationship between selected EOD sets and selected ECD sets. In two extremal cases we find that for the corresponding EOCD graphs G we have $\gamma_t(G) = \gamma(G)$ and $\gamma_t(G) = 2\gamma(G)$, respectively. In Section 3 we prove that the decision problem regarding whether a graph is an EOCD graph is an NP-complete problem. On the other hand, in one of the above extremal cases, EOCD graphs can be recognized in polynomial time. Then, in Section 4, we give a recursive description of EOCD trees, while in the final section EOCD graphs are characterized among the Sierpiński graphs.

We will use the notation $[n] = \{1, \ldots, n\}$ and $[n]_0 = \{0, \ldots, n-1\}$. Throughout the article we consider only finite, simple graphs. If S is a subset of vertices of a graph, then $\langle S \rangle$ denotes the subgraph induced by S. A *matching* of a graph is an independent set of its edges. For the later use we next state the following basic result. Its first assertion has been independently discovered several times, cf [19, Theorem 4.2], while for the second fact see [36].

Proposition 1.1. *Let G be a graph.*
(i) If P is an ECD set of G, then $|P| = \gamma(G)$.
(ii) If D is an EOD set of G, then $|D| = \gamma_t(G)$.

2. On the structure of EOCD graphs

In this section we first show that each EOD graph naturally yields an ECD graph and that each ECD graph can be modified to an EOD graph. Then we consider the structure of EOCD graphs with respect to the relationship between their EOD and ECD sets.

If D is an EOD set of a graph G, then D induces a matching M. Note that an edge from M lies in no triangle, hence its contracting produces no parallel edges. Now, let G' be the graph obtained from G by contraction all the edges from M. Then G' is an ECD graph with an ECD set consisting of the vertices obtained by the contraction of M.

Conversely, let G' be an ECD graph with an ECD set P. For every vertex $v \in P$ weakly partition the set of its neighbors arbitrarily into sets A and B. (If the degree of v is 1, then necessarily one of these sets is empty.) Let G be the graph obtained from G' by replacing every vertex $v \in P$ by two adjacent vertices v_A and v_B, and adding edges uv_A for every $u \in A$ and edges uv_B for every $u \in B$. Then G is an EOD graph with an EOD set $\{v_A, v_B : \ v \in P\}$.

Let G be an EOCD graph with an EOD set D and an ECD set P. Then $V(G)$ can be weakly partitioned into sets $D \cap P$, $D - P$, $P - D$, and $R = V(G) - (D \cup P)$, see Fig. 1. Clearly, some of these sets may be empty. From the definitions of ECD and EOD sets we infer the following properties.

- A vertex from $D \cap P$ (a black squared vertex in Fig. 1) can have an arbitrary number of neighbors in R, has a unique neighbor in $D - P$, and has no neighbors in $P - D$.

- A vertex from $P - D$ (a white squared vertex in Fig. 1) can have an arbitrary number of neighbors in R, a unique neighbor in $D - P$, and no neighbors in $D \cap P$.

- A vertex from $D - P$ (a black vertex in Fig. 1) can have an arbitrary number of neighbors in R and, either a unique neighbor in $P - D$ and a unique neighbor in $D - P$, or a unique neighbor in $D \cap P$.

- A vertex from R (a white vertex in Fig. 1) can have an arbitrary number of neighbors in R and either a unique neighbor in $P - D$ and a unique neighbor in $D - P$, or a unique neighbor in $D \cap P$.

- Vertices from $D \cap P$ together with their unique neighbors from $D - P$ induce a matching.

- Vertices from $P - D$ together with their unique neighbors in $D - P$ induce k copies of P_4, where $2k = |P - D|$.

The described structure is visible in Fig. 1. The same notation will be used later in Fig. 3.

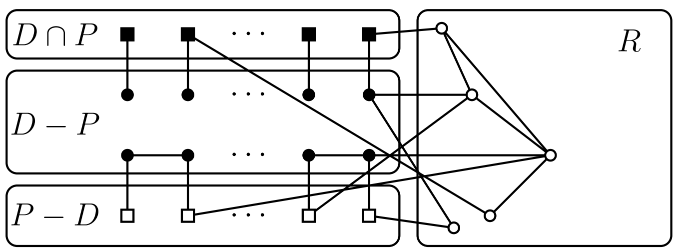

Figure 1: Structure of an EOCD graph.

The described structure above yields two extreme cases: either $D \cap P = \emptyset$ or $P - D = \emptyset$. Clearly different pairs of sets P, D in an EOCD graph can produce different configurations. In this sense, if there exists an ECD set P and an EOD set D in G, such that $D \cap P = \emptyset$, then we say that G is an EOCD graph *with empty* $D \cap P$, and if $P - D = \emptyset$, then we say that G is an EOCD graph *with empty* $P - D$. We observe that $D - P$ is always non-empty for every ECD set P and every EOD set D of any EOCD graph. Moreover, if $R = \emptyset$, then G is formed only from the disjoint union of copies of K_2 and copies of P_4.

The following two propositions follow directly from the above mentioned structure. The first result characterizes the EOCD graphs with empty $D \cap P$.

Proposition 2.1. *A graph G is an EOCD graph with empty $D \cap P$ if and only if there exists $A \subseteq V(G)$, such that $\langle A \rangle = kP_4$, where every vertex from $V(G) - A$ is adjacent to exactly one vertex of degree* 1 *in $\langle A \rangle$ and one vertex of degree* 2 *in $\langle A \rangle$.*

The second result characterizes the EOCD graphs with empty $P - D$.

Proposition 2.2. *A graph G is an EOCD graph with empty $P - D$ if and only if there exists $D \subseteq V(G)$ that induces a matching M, where every edge of M contains at least one vertex of degree 1 in G (this vertex is from $D - P$) and every vertex from $V(G) - D$ is adjacent to exactly one vertex in M which is in P.*

We end this section with a connection between $\gamma(G)$ and $\gamma_t(G)$ for EOCD graphs with empty $D \cap P$ or empty $P - D$, respectively. Both results follow from the described structure of EOCD graphs, and by applying Proposition 1.1.

Proposition 2.3. *If G is an EOCD graph with empty $D \cap P$, then $\gamma_t(G) = \gamma(G)$.*

Proposition 2.4. *If G is an EOCD graph with empty $P - D$, then $\gamma_t(G) = 2\gamma(G)$.*

Recall that for any graph G (without isolated vertices) $\gamma(G) \leq \gamma_t(G) \leq 2\gamma(G)$ holds. The above two propositions are of interest particularly because it is an open problem to characterize the graphs G with $\gamma_t(G) = 2\gamma(G)$, as well as the graphs G with $\gamma_t(G) = \gamma(G)$, cf. [21, p. 36]. In this direction, the trees T for which $\gamma_t(T) = \gamma(T)$ holds were characterized in [12, Theorem 6] as the trees obtained from a disjoint union of P_4s by means of four operations. Moreover, a characterization of trees T for which $\gamma_t(T) = 2\gamma(T)$ holds was obtained in [20]. For these two results see also [21, Sections 4.6 and 4.7].

3. Complexity results

In this section we deal with the problem of deciding whether a given graph contains an EOD set and an ECD set (EOCD Problem for short), that is, the following problem.

EOCD Problem	
Input:	A simple graph G.
Question:	Is G an EOCD graph?

In order to study this problem, we shall make a reduction from the one-in-three 3-SAT problem, which is known to be NP-complete [14] and reads as follows.

One-In-Three 3-SAT	
Input:	A Boolean formula $\mathcal{F}$ on n variables and m clauses.
Question:	Is there a satisfying truth assignment for the n variables, such that each clause has exactly one true literal?

Next we present the main result of this section, which is in part inspired by the proof of the NP-completeness of the problem of deciding whether a graph contains an EOD set given in [16].

Theorem 3.1. *The EOCD Problem is NP-complete.*

Proof. It is clear that the EOCD Problem is in NP, since verifying that a given set of vertices of a graph G is an EOD set or an ECD set can be done in polynomial time. We consider now a Boolean formula $\mathcal{F}$ with variables $X = \{x_1, \ldots, x_n\}$ and clauses $C = \{c_1, \ldots, c_m\}$. Each clause contains three literals, each of which we shall denote by x_i for a positive literal, or by $\overline{x_i}$ for a negative one. From the formula $\mathcal{F}$, we construct a graph G_F in the following way. For any variable $x_i \in X$, add to G_F the graph G_i from Fig. 2. For each clause $c_i \in C$, we add a vertex y_i. Now, if a variable x_i occurs as a positive literal in a clause c_j, then add the edge $y_j u_i$, otherwise (if a variable x_i occurs as a negative literal in a clause c_j) add the edge $y_j \overline{u_i}$. Clearly, G_F can be constructed in polynomial time.

We claim that G_F is an EOCD graph if and only if there is a satisfying truth assignment for the n variables in the Boolean formula $\mathcal{F}$, such that each clause has exactly one true literal, that is, if and only if $\mathcal{F}$ has a one-in-three satisfying truth assignment.

Assume first that $\mathcal{F}$ has a one-in-three satisfying truth assignment. We construct two sets D and P in the following way. Add to D the vertices $q_i, c_{i1}, c_{i4}, c_{i5}$, and to P the vertices q_i, c_{i3}, c_{i6} for every $i \in [n]$. Now, if the variable x_i is assigned the value true, then we add to D the vertices $u_i, v_{i1}, w_{i4}, w_{i5}$, and to P the vertices u_i, w_{i3}, w_{i7}. On the other hand, if x_i is assigned the value false, then we add to D the vertices $\overline{u_i}, v_{i2}, w_{i3}, w_{i4}$, and to P the vertices $\overline{u_i}, w_{i2}, w_{i5}$. It is easy to see that $D \cap V(G_i)$ is an EOD set and $P \cap V(G_i)$ an ECD set of G_i. Moreover, since the truth assignment has

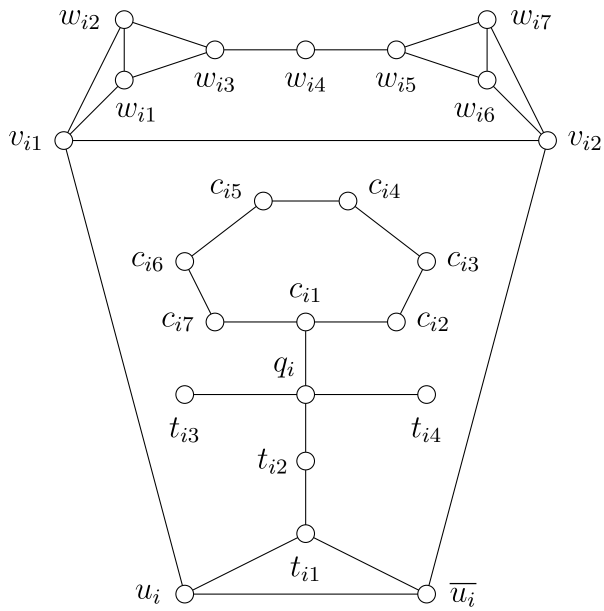

Figure 2: The graph G_i corresponding to a variable x_i.

exactly one literal with value true, each vertex y_j, with $j \in [m]$, is adjacent to exactly one vertex of D and exactly one vertex of P (clearly both vertices coincide). Thus, D is an EOD set and P is an ECD set in G_F and, as a consequence, G_F is an EOCD graph.

Conversely, assume that G_F is an EOCD graph. Let D be an EOD set and P an ECD set in G_F. We next collect several facts regarding the sets D and P.

- The vertex q_i ($i \in [n]$) lies in $D \cap P$. Indeed, this fact follows because q_i is adjacent to leaves t_{i3} and t_{i4}.
- The vertices c_{i1}, c_{i4}, c_{i5} ($i \in [n]$) belong to D. The vertex c_{i1} belongs to D because otherwise the 7-cycle on vertices c_{ij} cannot be efficiently open dominated. We then consequently see that also $c_{i4}, c_{i5} \in D$.
- The vertices $t_{i2}, t_{i3}, t_{i4}, c_{i1}$ ($i \in [n]$) do not lie in P, and the vertices c_{i3}, c_{i6} ($i \in [n]$) lie in P. These facts follow immediately from the first point.
- Either ($u_i \notin D$ and $\overline{u_i} \in D$) or ($u_i \in D$ and $\overline{u_i} \notin D$). Similarly, either

($u_i \notin P$ and $\overline{u_i} \in P$) or ($u_i \in P$ and $\overline{u_i} \notin P$). Indeed, since $q_i \in D \cap P$ and $c_{i1} \in D$, for every $i \in [n]$, the vertices $t_{i1}, t_{i2} \notin D \cup P$. Thus, t_{i1} must be dominated either by u_i or by $\overline{u_i}$ in D and in P.

- If $u_i \in P$ and $\overline{u_i} \notin P$, then $v_{i1} \notin P$ and every vertex y_j such that the variable x_i belongs to the clause c_j does not belong to P. Moreover, to efficiently dominate the vertices $v_{i2}, w_{i1}, \ldots, w_{i7}$ we clearly have that $w_{i3} \in P$ and exactly one vertex of the pair w_{i6}, w_{i7} belongs to P.

- Analogously, if $u_i \notin P$ and $\overline{u_i} \in P$, then we obtain that $v_{i2} \notin P$ and every vertex y_j such that the variable x_i belongs to the clause c_j does not belong to P. Also, $w_{i5} \in P$ and exactly one vertex of the pair w_{i1}, w_{i2} belongs to P.

- If $u_i \in D$ and $\overline{u_i} \notin D$, then either $v_{i1} \in D$ or there exists a vertex $y_j \in D$ such that the variable x_i appears as positive in the clause c_j. If the latter happens ($y_j \in D$), then $v_{i1} \notin D$. It is straightforward to observe that, in such a case, any subset of vertices of the set $\{v_{i2}, w_{i1}, \ldots, w_{i7}\}$ does not efficiently open dominate the same set of vertices $\{v_{i2}, w_{i1}, \ldots, w_{i7}\}$, which is a contradiction. Thus, $y_j \notin D$ and therefore $v_{i1} \in D$. We also observe that $w_{i4}, w_{i5} \in D$.

- Analogously to the last item, if $u_i \notin D$ and $\overline{u_i} \in D$, then $v_{i2}, w_{i3}, w_{i4} \in D$.

As a consequence of the above facts, we have that either $u_i, v_{i1} \in D$ or $\overline{u_i}, v_{i2} \in D$, and either $u_i \in P$ or $\overline{u_i} \in P$. Now, we say that a subgraph G_i of G_F, corresponding to a variable x_i, is *nice* if either $u_i \in D \cap P$ or $\overline{u_i} \in D \cap P$. Assume that there exists G_i which is not nice, *i.e.*, $u_i, \overline{u_i} \notin D \cap P$. Hence, either $u_i \in D$ and $\overline{u_i} \in P$ or $u_i \in P$ and $\overline{u_i} \in D$. Consider a clause c_j such that $x_i \in c_j$. Hence, y_j is dominated either by u_i or by $\overline{u_i}$ from G_i, which means either by D or by P, say D. Therefore, there must exist another variable $x_\ell \in c_j$, such that y_j is dominated also by P and not by D. This implies that G_ℓ is not nice as well. Notice that for the third literal $x_k \in c_j$, G_k must be nice. In general, for every clause c_j, either all three corresponding graphs are nice, or exactly one is nice and two are not. Moreover, if the later is true, then y_j is not dominated from D and from P by the nice subgraph.

Let $P_i = P \cap V(G_i)$ ($i \in [n]$). For every not nice graph G_i we exchange some vertices of P_i as follows. If $u_i \in P_i$, then $P'_i = (P_i - \{u_i, w_{i3}, w_{i6}, w_{i7}\}) \cup$

$\{\overline{u_i}, w_{i2}, w_{i5}\}$, and if $\overline{u_i} \in P$, then $P'_i = (P_i - \{\overline{u_i}, w_{i5}, w_{i1}, w_{i2}\}) \cup \{u_i, w_{i3}, w_{i7}\}$. If G_i is nice, then $P'_i = P_i$. We claim that $P' = \cup_{i=1}^n P'_i$ is an ECD set, such that together with D every subgraph G_i is nice. Clearly P'_i is an ECD set for G_i by the items above. If some y_j was dominated by a vertex u_i (or by $\overline{u_i}$) which was in P_i but not now in P'_i, then y_j is now dominated either by u_ℓ or by $\overline{u_\ell}$, where x_i and x_ℓ are those variables from the clause c_j, for which G_i and G_ℓ were not nice. Thus, P' is an ECD set. Moreover, the EOD set D and the ECD set P' lead to the fact that every G_i is nice. Since D is an EOD set and P' is an ECD set, then every vertex y_j corresponding to a clause is adjacent to exactly one vertex u_i or $\overline{u_i}$ of G_i. Now, if $u_i \in D \cap P'$, then we set the variable x_i as true, otherwise (if $\overline{u_i} \in D \cap P'$) set x_i as false. It clearly follows that such an assignment is a truth assignment in exactly one literal in every clause for $\mathcal{F}$ and the proof is complete. $\square$

In view of Theorem 3.1, it is reasonable to try to find some special classes of graphs for which the EOCD Problem is polynomial. Simple examples are provided by the paths P_n which are EOCD graphs if and only if $n \not\equiv 1 \pmod 4$ and the cycles C_n which are EOCD graphs if and only if $n \equiv 0 \pmod{12}$. Note also that a complete bipartite graph $K_{r,t}$ is an EOCD graph if and only if $r = 1$ or $t = 1$. Moreover, the hypercube Q_n is an EOCD graph if and only if $n = 1$. Indeed, suppose that Q_n, $n \geq 1$, is an EOCD graph. As Q_n is n-regular, its order must be divisible by n (because it admits an EOD set) as well as by $n + 1$ (since it admits an ECD set). Since the order of Q_n is 2^n, this is only possible if $n = 1$.

We end this section with a discussion on extreme cases with respect to the structure of EOCD graphs as described in Section 2.

Theorem 3.2. *If G is a graph on n vertices and m edges, then it can be decided in $O(nm)$ time whether G is an EOCD graph with empty $P - D$.*

Proof. Let G be a graph. Clearly, components which are isomorphic to K_2 (if they exist) do not influence the fact that G is an EOCD graph or not. Hence we may restrict our attention to the case when G has no components isomorphic to K_2. If there exists no degree 1 vertex, then by Proposition 2.2, G is not an EOCD graph with empty $P - D$. Let P be the set of all support vertices of degree one vertices. For every support from P choose exactly one neighbor of degree 1 and let D be a set containing P as well as the chosen vertices of degree 1. By Proposition 2.2 one only needs to check if D and P are an EOD set and an ECD set of G, respectively. Even more, it is clear that P is an ECD set in G if and only if D is an EOD set of G. Hence it is enough to check whether the union of closed neighborhoods centered at P covers $V(G)$ and whether these closed neighborhoods have pairwise empty intersection. The first task can be clearly done in $O(m)$ time. For the second task it suffices to check if the distance between any two different vertices from P is at least 3. Clearly, this can be done in time $O(mn)$, if we start the BFS algorithm in an arbitrary vertex of P. □

We end the section with a question about the other extremal case.

Problem 3.3. *Can it be checked in polynomial time whether G is an EOCD graph with empty $D \cap P$?*

4. EOCD trees

Let T' be an EOCD tree with an EOD set D' and an ECD set P'. We now introduce five operations that construct larger EOCD trees from T'. In the main theorem of this section we will then prove that these operations are characteristic for EOCD trees. The operations are illustrated in Fig. 3 where we use the convention introduced in Section 2: a vertex from $D \cap P$ is black squared, a vertex from $P - D$ is white squared, a vertex from $D - P$ is black circled, and the remaining vertices are white circled.

(O_1) For $u \in D' \cap P'$ we obtain T from T' by adding a vertex v and edge uv.

(O_2) For $w \notin D'$ we obtain T from T' by adding a path xuv and edge wx.

(O_3) For $t \in D' - P'$ we obtain T from T' by adding a path $zwxuv$ and edge tz.

(O_4) For a path vux with $\deg(v) = 1$, $\deg(u) = 2$, $u, x \in D'$, and $u \in P'$, we obtain T from T' by adding a vertex y and edge yx.

(O_5) For a path $uxwzw'x'$ with $\deg(u) = \deg(x') = 1$, $\deg(x) = \deg(w) = \deg(w') = 2$, $u, x, w', x' \in D'$, and $x, w' \in P'$, we obtain T from T' by adding a vertex v and edge uv.

The main difference between these five operations is that for O_1 the original EOD set and ECD set do not change, for O_2 and O_3 we add some vertices to the EOD set and to the ECD set, while for O_4 and O_5 the EOD set remains the same and we exchange some vertices in the ECD set.

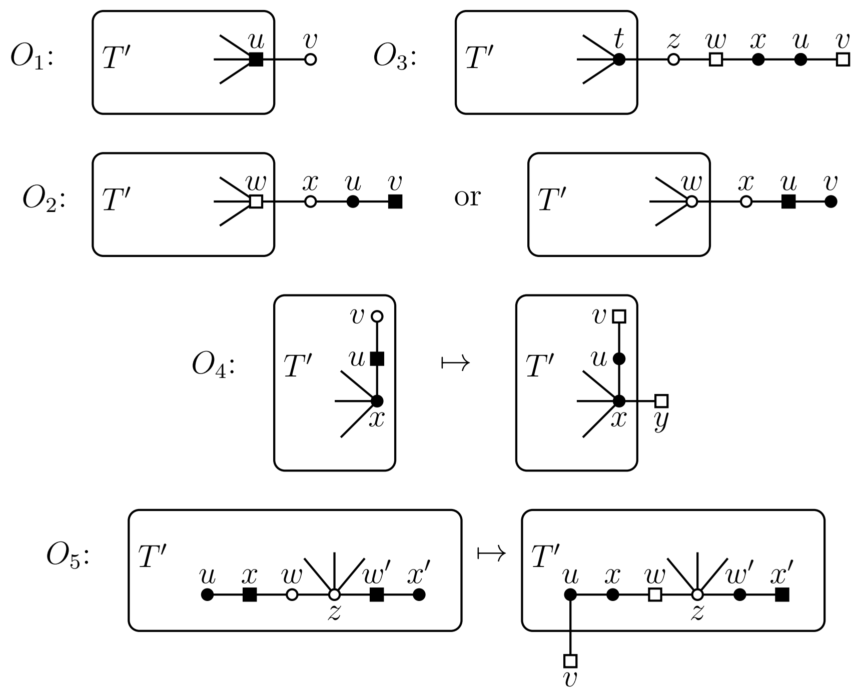

Figure 3: Operations $O_1 - O_5$.

Theorem 4.1. *A tree T is an EOCD graph if and only if T can be obtained from K_2 by a sequence of operations $O_1 - O_5$.*

Proof. Assume first that T is a tree obtained from K_2 by a sequence of operations $O_1 - O_5$. We will show that T is an EOCD tree by induction on the length k of the mentioned sequence. If $k = 0$, then $T \cong K_2$ which is an EOCD graph. Let now $k > 0$ and let T' be a tree obtained from K_2 by using the same sequence as for T, but without including the last step. By the induction hypothesis, T' is an EOCD tree with an EOD set D' and an ECD set P'. If T is obtained from T' by operation O_1, then clearly T is an EOCD tree for $D = D'$ and $P = P'$ (see the upper left diagram of Fig. 3). If T is obtained from T' by operation O_2, then T is an EOCD tree where $D = D' \cup \{u, v\}$. The set P depends whether w is in P' or not. If $w \in P'$, then $P = P' \cup \{v\}$, and if $w \notin P'$, then $P = P' \cup \{u\}$ (see the diagrams of the second line of Fig. 3). Suppose now that we apply operation O_3 on T' to get T. Again it is straightforward to see that T is an EOCD graph for $D = D' \cup \{u, x\}$ and $P = P' \cup \{v, w\}$ (see the upper right diagram of Fig. 3). If operation O_4 is applied to get T from T', then we set $D = D'$ and $P = (P' - \{u\}) \cup \{v, y\}$ and T is an EOCD tree again (see the diagram in the third line of Fig. 3). Finally, if T is obtained from T' by operation O_5, then it is not hard to see that T is an EOCD tree for $D = D'$ and $P = (P' - \{x, w'\}) \cup \{v, x', w\}$ (see the lower diagram of Fig. 3).

To prove the converse, let T be an EOCD tree with an EOD set D and an ECD set P. Let $r \in V(T)$ be a vertex of T and consider T as a rooted tree with the root r. Let v be a vertex of degree 1 of T that is at the maximum distance from r and let u be the support vertex of v. Clearly $u \in D$, while either $u \in P$ or $v \in P$. We call a neighbor y of x a *down-* (resp. *up-*) *neighbor* of x if y is further (resp. closer) from r than x. We proceed by induction on the number of vertices of T. Clearly, K_2 is the smallest EOCD tree, hence the base of the induction. We distinguish the following cases.

Case 1: $v \notin P$ and $v \notin D$.
In this case $u \in P \cap D$. We obtain a tree T' from T by deleting v. Clearly T' is an EOCD tree with $D' = D$ and $P' = P$. By the induction hypothesis T' can be built from K_2 by a sequence of operations $O_1 - O_5$. If we add the operation O_1 at the end of this sequence, then we obtain T from K_2 by a sequence of operations $O_1 - O_5$.

Case 2: $v \notin P$ and $v \in D$.
Then $u \in P \cap D$. If $\deg(u) = 1$, then $T \cong K_2$ and we are done. So, let $\deg(u) > 1$. If u is the support for more degree 1 vertices than v, then we have Case 1. (Notice that the same occurs when $u = r$.) Thus let $\deg(u) = 2$. Let x be the up-neighbor of u. If $\deg(x) > 2$, then x has a down-neighbor y different from u. If $\deg(y) = 1$, then we have a contradiction with P being an ECD set of T, since $u \in P$ implies that y and x cannot be in P and therefore y is neither dominated by P nor $y \in P$. So $\deg(y) > 1$ and let y' be a down-neighbor of y. Clearly, $\deg(y') = 1$ by the choice of v. This yields a contradiction with D being an EOD set of T, since y cannot be in D because x is already dominated by $u \in D$. Thus, $\deg(x) = 2$ and let w be the up-neighbor of x or the other down-neighbor when $x = r$. By the choice of v, x must be different from r or we obtain the same problems as for $\deg(x) > 2$. Since x is the neighbor of $u \in D \cap P$, we have that $w \notin D$ and $w \notin P$. Let T' be the tree obtained from T by deleting vertices v, u, x. Then T' is an EOCD tree with $D' = D - \{u, v\}$ and $P' = P - \{u\}$. By the induction hypothesis, T' can be built from K_2 by a sequence of operations

$O_1 - O_5$. Adding operation O_2 at the end of this sequence we obtain T from K_2 by a sequence of operations $O_1 - O_5$.

Case 3: $v \in P \cap D$.
If $\deg(u) = 1$, then $T \cong K_2$ and we are done. If $\deg(u) > 2$, then we have a contradiction with P being an ECD set. Thus $\deg(u) = 2$. Also notice that $u \neq r$, since otherwise T would be isomorphic to P_3, which is not possible with $v \in P$. Let x be the up-neighbor of u. Clearly $x \notin D \cup P$. If x had a down-neighbor different from u or if $x = r$, then we have a contradiction with D being an EOD set of T. Thus, also $\deg(x) = 2$ and let w be the up-neighbor of x. Notice that now w must be in P to dominate x, but again $w \notin D$. Let T' be the tree obtained from T by deleting vertices v, u, x. Clearly T' is an EOCD tree with $D' = D - \{u, v\}$ and $P' = P - \{v\}$. By the induction hypothesis T' can be built from K_2 by a sequence of operations $O_1 - O_5$. Adding the corresponding operation O_2 at the end of this sequence we obtain T from K_2 as desired.

Case 4: $v \in P$ and $v \notin D$.
In this case $u \notin P$. If $u = r$, then we have a contradiction with P being an ECD set when $\deg(u) > 1$ and with D being an OED set if $\deg(u) = 1$. So we may assume that $u \neq r$. Clearly $\deg(u) = 2$, otherwise we have a contradiction again with P being an ECD set of T and by the choice of v. Let x be the up-neighbor of u. Since $v \notin D$ and $v \in P$, we have that $x \in D$ and $x \notin P$, respectively. The only second down-neighbor of x is v, otherwise we have a contradiction with D being an EOD set for T according to the choice of v. Suppose that x has a down-neighbor y of degree 1. Clearly, $v \in P$ implies $x \notin P$ and therefore $y \in P$ and y is the unique down-neighbor of x of degree 1. Thus $vuxy$ is a path. Let T' be a tree obtained from T by deleting the vertex y. Clearly T' is an EOCD tree with $D' = D$ and $P' = (P - \{v, y\}) \cup \{u\}$. By the induction hypothesis T' can be built from K_2 by a sequence of operations $O_1 - O_5$. If we add the operation O_4 at the end of this sequence, then we obtain T from K_2 by a sequence of operations $O_1 - O_5$.

Suppose now that x has no down-neighbor of degree 1. If $x = r$, then we have a contradiction with P being an ECD set for T. Hence $x \neq r$ and $\deg(x) = 2$ holds. Let w be the up-neighbor of x. Clearly, $w \in P$ and $w \notin D$. If $\deg(w) \geq 3$, then w has a down-neighbor x' other than x. To dominate x' from D, the vertex x' must have a down-neighbor u' which is in D and the same holds for u', which must have a down-neighbor v' which is also in

D. Moreover, to dominate x', u' and v' from P exactly once, also $v' \in P$ holds. Notice that $\deg(x') = 2$ according to that D is an EOD set of T, and $\deg(u') = 2$ since P is an ECD set of T. The situation for v' is now as in Case 3 and we are done if $\deg(w) \geq 3$.

Thus, from now on, we consider $\deg(w) = 2$ and let z be the up-neighbor of w (or down-neighbor if $w = r$). Again $z \notin P$ since $w \in P$, and $z \notin D$ since $x \in D$. We consider the following subcases.

Subcase 4.1: $\deg(z) \geq 3$.
Let $w' \neq w$ be a down-neighbor of z. Since z is dominated from P by w, w' is not in P and therefore, w' must have a down-neighbor x' which is in P. Also, x' cannot have a second down-neighbor by the choice of v and the structure of P. We consider two possibilities regarding the vertex w'.

Subcase 4.1.1: $w' \notin D$.
In this subcase w' must be dominated by a down-neighbor in D. If $x' \notin D$, then we have a contradiction since x' has no second-down neighbor and D is an EOD set. Hence, $x' \in D$ and x' must have a down-neighbor $u' \in D$ for x' to be dominated by D. Clearly $u' \notin P$. If x' has another down-neighbor u'', then we obtain a tree T' from T by deleting v. Clearly T' is an EOCD tree with $D' = D$ and $P' = P$. By the induction hypothesis, T' can be built from K_2 by a sequence of operations $O_1 - O_5$. If we add the operation O_1 at the end of this sequence, then we obtain T from K_2 by a sequence of operations $O_1 - O_5$. So we may assume that $\deg(x') = 2$. If $\deg(w') = 2$, then let T' be a tree obtained from T by deleting vertices u', x', w'. Clearly T' is an EOCD tree with $D' = D - \{u', x'\}$ and $P' = P - \{x'\}$. By the induction hypothesis, T' can be built from K_2 by a sequence of operations $O_1 - O_5$. Adding the corresponding operation O_2 at the end of this sequence we obtain T from K_2 as desired. On the other hand, if w' has a down-neighbor x'' other than x', then x'' is not in D and not in P, since w' is already dominated by x' in both P and D. Moreover, x'' must be dominated by its down-neighbor u'' in both P and D. Furthermore, u'' is dominated by its down-neighbor v'' in D. If $\deg(u'') > 2$, then we have Case 1. So let $\deg(u'') = 2$. If $\deg(x'') > 2$, we have a contradiction with D being an EOD set of T or by the choice of v. Hence, $\deg(x'') = 2$ and we proceed like in Case 2 for v'', u'', x''.

Subcase 4.1.2: $w' \in D$.
Since $z \notin D$, also in this subcase w' must have a down-neighbor in D. Suppose first that $x' \in D$ (and recall that $x' \in P$). If x' has a down-neighbor u', then $\deg(u') = 1$ by the choice of v and since P is an ECD set. Let T'

be a tree obtained from T by deleting u'. Clearly T' is an EOCD tree with $D' = D$ and $P' = P$. By the induction hypothesis T' can be built from K_2 by a sequence of operations O_1-O_5, and attaching operation O_1 to this sequence we obtain T from K_2 as desired. Thus, we may assume that $\deg(x') = 1$. Observe that $\deg(w') = 2$, otherwise we have a contradiction with the choice of v, since P is an ECD set and since D is an EOD set. Let T' be a tree obtained from T by deleting v. Clearly T' is an EOCD tree with $D' = D$ and $P' = (P - \{x', w, v\}) \cup \{x, w'\}$. By the induction hypothesis, T' can be built from K_2 by a sequence of operations $O_1 - O_5$. Now, adding operation O_5 at the end of such sequence produces our desired result. Next, let $x' \notin D$. Clearly $\delta(x') = 1$, since D is an EOD set and by the choice of v. Let w now be dominated by $x'' \in D$. To dominate x'' from P, let u'' be its down-neighbor. Also, $\delta(u'') = 1$ since D is an EOD set and by the choice of v. Observe that $\delta(x'') = 2$ since any other down-neighbor u''' of x'' would required a down-neighbor v' in P, which is not possible since D is an EOD set and by the choice of v. Let T' be a tree obtained from T by deleting vertex x'. Clearly T' is an EOCD tree with $D' = D$ and $P' = (P - \{x', u''\}) \cup \{x''\}$. By the induction hypothesis, T' can be built from K_2 by a sequence of operations $O_1 - O_5$. If we add operation O_4 at the end of this sequence, then we obtain T from K_2 by a sequence of operations $O_1 - O_5$.

Subcase 4.2: $\deg(z) = 2$.
Let T' be a tree obtained from T by deleting v, u, x, w, z. Clearly T' is an EOCD tree with $D' = D - \{u, x\}$ and $P' = P - \{v, w\}$. Applying the induction hypothesis once more and ending with an additional operation O_3, we again obtain T from K_2 as desired and we are done. $\square$

It is not obvious that all the five operations are necessary to characterize EOCD trees. To see that this is the case, note first that P_3 can be obtained from K_2 only by operation O_1 and that the sequence of operations O_1, O_4 is unique for P_4. Similarly, the sequence of operations O_1, O_2 is unique for P_6. To infer that operations O_3 and O_5 are also indispensable, consider the following more elaborate examples.

Let T be the tree obtained from $K_{1,3}$ by subdividing one of its edges with five vertices and each of the other two edges with eight vertices. A short analysis reveals that T is an EOCD tree where the vertex of degree 3 must be in $D \cap P$ and that its neighbor on the shortest leg must be in D. After this observation, operation O_3 cannot be avoided when constructing T in view of Theorem 4.1. For operation O_5, let P_{22}^{+} be the graph obtained from the path on 22 vertices $v_1, \ldots, v_{22}$, by adding vertices u, w, x, y and edges $v_5u, uw, v_{18}x, xy$. One can observe that P_{22}^+ is an EOCD tree with a unique EOD set D and a unique ECD set P. From here it can be concluded that operation O_5 is needed to build P_{22}^+ from K_2 in view of Theorem 4.1. We leave the details to the reader.

5. EOCD Sierpiński graphs

The Sierpiński graphs S_p^n were introduced in [26] and afterwards investigated from many different aspects. Here we only mention recent studies of Sierpiński graphs related to codes and domination [11, 17, 37], their shortest paths [22, 43], and an appealing generalization of Sierpiński graphs due to Hasunuma [18] that in turn extends several known results about Sierpiński graphs. For the additional vast bibliography on these graphs we refer to [18].

The *Sierpiński graphs* S_p^n, $p \geq 1$, $n \geq 0$, are defined as follows. $S_p^0 = K_1$ for any p. For $n \geq 1$, the vertex set of S_p^n is $[p]_0^n$, we shall denote its elements by $s = s_n \ldots s_1$. Vertices $s_n \ldots s_1$ and $t_n \ldots t_1$ are adjacent if and only if there exists a $\delta \in [n]$ such that

(i) $s_d = t_d$, for $d \in [n] - [\delta]$;

(ii) $s_\delta \neq t_\delta$; and

(iii) $s_d = t_\delta$ and $t_d = s_\delta$ for $d \in [\delta - 1]$.

Note that $S_1^n \cong K_1$ ($n \geq 1$), $S_2^n \cong P_{2^n}$ ($n \geq 1$), and $S_p^1 \cong K_p$ ($p \geq 1$). Hence, for our purposes we may restrict the attention to the Sierpiński graphs S_p^n with $p \geq 3$ and $n \geq 2$.

The edge set of S_p^n can be equivalently defined recursively as

$$E(S_p^n) = \{\{is, it\} : i \in [p]_0 \, , \{s, t\} \in E(S_p^{n-1})\} \cup \{\{ij^{n-1}, ji^{n-1}\} \mid i, j \in [p]_0 \, , i \neq j\} \, .$$

This implies that S_p^n can be constructed from p copies of S_p^{n-1} as follows. For each $j \in [p]_0$ concatenate j to the left of the vertices in a copy of S_p^{n-1} and denote the obtained graph with jS_p^{n-1}. Then for each $i \neq j$ join copies iS_p^{n-1} and jS_p^{n-1} by the single edge $e_{ij}^{(n)} = \{ij^{n-1}, ji^{n-1}\}$.

If $1 \leq d < n$ and $\underline{s} \in [p]_0^d$, then the subgraph of S_p^n induced by the vertices whose labels begin with $\underline{s}$ is isomorphic to S_p^{n-d}. It is denoted with $\underline{s}S_p^{n-d}$ in accordance with the above notation jS_p^{n-1}. Note that S_p^n contains p^d pairwise disjoint subgraphs $\underline{s}S_p^{n-d}$, $\underline{s} \in [p]_0^d$. In particular, S_p^n contains p^{n-1} pairwise disjoint p-cliques $\underline{s}S_p^1$, $\underline{s} \in [p]_0^{n-1}$. The vertices i^n, $i \in [p]_0$, of S_p^n are called *extreme vertices* (of S_p^n). The clique in which an extreme vertex lies is called an *extreme clique*.

After this preparation we can state the following result which asserts, roughly speaking, that precisely one half of the Sierpiński graphs are EOCD graphs.

Theorem 5.1. *Let $p \geq 3$ and $n \geq 2$. Then S_p^n is an EOCD graph if and only if p is even.*

Proof. Suppose that p is odd and that D is an EOD set of S_p^n. Observe first that no extreme vertex of S_p^n lies in D because otherwise D would contain two vertices from the same extreme clique, which is not possible. Hence every vertex of D is of degree p and consequently $|D| = |V(S_p^n)|/p = p^{n-1}$. Since this is at the same time the number of all p-cliques of S_p^n, it follows that D must have precisely one vertex in each p-clique of S_p^n. By the same argument as above, a vertex s of D can only be covered by a vertex t of D that lies in a p-clique that is neighboring the p-clique of s. This means that the vertices of D can be partitioned into disjoint pairs. But p is odd and hence $|D| = p^{n-1}$ is odd as well, hence D does not exist.

Assume now that p is even, say $p = 2k$, $k \geq 2$. We first recall from [27] that S_p^n contains an ECD set. In order to prove that S_p^n is an EOCD graph it thus remains to prove that it contains an EOD set. For this sake set

$$D_{2i} = \{\underline{s}(2i)(2i+1) : \ \underline{s} \in [p]_0^{n-2}\}, \quad 0 \leq i \leq k-1\,,$$

and

$$D_{2i+1} = \{\underline{s}(2i+1)(2i) : \ \underline{s} \in [p]_0^{n-2}\}, \quad 0 \leq i \leq k-1\,.$$

We claim that

$$D = \bigcup_{i=0}^{2k-1} D_i$$

is an EOD set of S_p^n. Note first that for any $i \in [k]_0$, $|D_{2i}| = |D_{2i+1}| = p^{n-2}$. Since the sets D_i, $i \in [2k]_0$, are clearly pairwise disjoint, it follows that $|D| = 2kp^{n-2} = p^{n-1}$. Let now $\underline{s}S_p^1$, $\underline{s} = s_n \ldots s_2 \in [p]_0^{n-1}$, be an arbitrary p-clique of S_p^n. If s_2 is even, say $s_2 = 2i$, then $\underline{s}(2i+1) \in D \cap \underline{s}S_p^1$, and if s_2 is odd, say $s_2 = 2i+1$, then $\underline{s}(2i) \in D \cap \underline{s}S_p^1$. If follows that any p-clique contains a vertex of D and consequently it contains exactly one such vertex. Since by the construction of the sets D_{2i} and D_{2i+1} any vertex of D has a neighbor in D, we conclude that D is indeed an EOD set of S_p^n. $\square$

Combining the construction of the EOC sets in the proof of Theorem 5.1 with Proposition 1.1(ii) we get:

Corollary 5.2. *If $p \geq 4$ is even and $n \geq 2$, then $\gamma_t(S_p^n) = p^{n-1}$.*

[1] G. Abay-Asmerom, R. H. Hammack, D. T. Taylor, Total perfect codes in tensor products of graphs, Ars Combin. 88 (2008) 129–134.

[2] G. Abay-Asmerom, R. H. Hammack, D. T. Taylor, Perfect r-codes in strong products of graphs, Bull. Inst. Combin. Appl. 55 (2009) 66–72.

[3] D. W. Bange, A. E. Barkauskas, P. J. Slater, Efficient dominating sets in graphs. (In: Applications of Discrete Mathematics, Clemson, 1986), SIAM, Philadelphia, 1988, 189–199.

[4] A. Brandstädt, P. Fičur, A. Leitert, M. Milanič, Polynomial-time algorithms for weighted efficient domination problems in AT-free graphs and dually chordal graphs, Inform. Process. Lett. 115 (2015) 256–262.

[5] N. Biggs, Perfect codes in graphs, J. Combin. Theory Ser. B 15 (1973) 289–296.

[6] B. Brešar, P. Dorbec, W. Goddard, B. Hartnell, M. A. Henning, S. Klavžar, D. F. Rall, Vizing's conjecture: a survey and recent results, J. Graph Theory 69 (2012) 46–76.

[7] R. Cowen, S. H. Hechler, J. W. Kennedy, A. Steinberg, Odd neighborhood transversals on grid graphs, Discrete Math. 307 (2007) 2200–2208.

[8] P. Cull, I. Nelson, Error-correcting codes on the towers of Hanoi graphs, Discrete Math. 208-209 (1999) 157–175.

[9] I. J. Dejter, Perfect domination in regular grid graphs, Australas. J. Combin. 42 (2008) 99–114.

[10] Y.-P. Deng, Efficient dominating sets in circulant graphs with domination number prime, Inform. Process. Lett. 114 (2014) 700–702.

[11] P. Dorbec, S. Klavžar, Generalized power domination: propagation radius and Sierpiński graphs, Acta Appl. Math. 134 (2014) 75–86.

[12] M. Dorfling, W. Goddard, M. A. Henning, C. M. Mynhardt, Construction of trees and graphs with equal domination parameters, Discrete Math. 306 (2006) 2647–2654.

[13] M. R. Fellows, M. N. Hoover, Perfect domination, Australas. J. Combin. 3 (1991) 141–150.

[14] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., New York, 1979.

[15] H. Gavlas, K. Schultz, Efficient open domination, Electron. Notes Discrete Math. 11 (2002) 681–691.

[16] H. Gavlas, K. Schultz, P. Slater, Efficient open domination in graphs, Sci. Ser. A Math. Sci. (N.S.) 6 (2003) 77–84.

[17] S. Gravier, M. Kovše, M. Mollard, J. Moncel, A. Parreau, New results on variants of covering codes in Sierpiński graphs, Des. Codes Cryptogr. 69 (2013) 181–188.

[18] T. Hasunuma, Structural properties of subdivided-line graphs, J. Discrete Algorithms 31 (2015) 69–86.

[19] T. W. Haynes, S. T. Hedetniemi, P. J. Slater, Fundamentals of Domination in Graphs, Marcel Dekker, New York, 1998.

[20] M. A. Henning, Trees with large total domination number, Util. Math. 60 (2001) 99–106.

[21] M. A. Henning, A Yeo, Total Domination in Graphs, Springer, New York, 2013.

[22] A.M. Hinz, C. Holz auf der Heide, An efficient algorithm to determine all shortest paths in Sierpiński graphs, Discrete Appl. Math. 177 (2014) 111–120.

[23] I. Hrastnik Ladinek, J. Žerovnik, Perfect codes in direct graph bundles, Inform. Process. Lett. 115 (2015) 707–711.

[24] P. K. Jha, Tight-optimal circulants vis-à-vis twisted tori, Discrete Appl. Math. 175 (2014) 24–34.

[25] P. K. Jha, 1-perfect codes over dual-cubes vis-à-vis Hamming codes over hypercubes, IEEE Trans. Inform. Theory 61 (2015) 4259–4268.

[26] S. Klavžar, U. Milutinović, Graphs $S(n, k)$ and a variant of the Tower of Hanoi problem, Czechoslovak Math. J. 47(122) (1997) 95–104.

[27] S. Klavžar, U. Milutinović, C. Petr, 1-perfect codes in Sierpiński graphs, Bull. Austral. Math. Soc. 66 (2002) 369–384.

[28] S. Klavžar, S. Špacapan, J. Žerovnik, An almost complete description of perfect codes in direct products of cycles, Adv. in Appl. Math. 37 (2006) 2–18.

[29] W. F. Klostermeyer, E. M. Eschen, Perfect codes and independent dominating sets, Congr. Numer. 142 (2000) 7–28.

[30] W. F. Klostermeyer, J. L. Goldwasser, Total perfect codes in grid graphs, Bull. Inst. Combin. Appl. 46 (2006) 61–68.

[31] M. Knor, P. Potočnik, Efficient domination in cubic vertex-transitive graphs, European J. Combin. 33 (2012) 1755–1764.

[32] T. Kraner Šumenjak, I. Peterin, D. F. Rall, A. Tepeh, Partitioning the vertex set of G to make $G \,\Box\, H$ an efficient open domination graph, Discrete Math. Theoret. Comput. Sci. 18 (2016) #1503.

[33] J. Kratochvíl, Perfect codes of graphs, J. Combin. Theory Ser. B 40 (1986) 224–228.

[34] J. Kratochvíl, M. Křivánek, On the computational complexity of codes in graphs. (In: Mathematical Foundations of Computer Science, Carlsbad, 1988) Lecture Notes in Comput. Sci. 324, Springer, Berlin, 1988, 396–404.

[35] J. Kratochvíl, P. Manuel, M. Miller, Generalized domination in chordal graphs, Nordic J. Comput. 2 (1995) 41–50.

[36] D. Kuziak, I. Peterin, I. G. Yero. Efficient open domination in graph products, Discrete Math. Theoret. Comput. Sci. 16 (2014) 105–120.

[37] C.-H. Lin, J.-J. Liu, Y.-L. Wang, Global strong defensive alliances of Sierpiński-like graphs, Theory Comput. Syst. 53 (2013) 365–385.

[38] A. McRae, Generalizing NP-completeness Proofs for Bipartite Graphs and Chordal Graphs, PhD Thesis, Clemson University, 1994.

[39] M. Mollard, On perfect codes in Cartesian products of graphs, European J. Combin. 32 (2011) 398–403.

[40] C. B. Smart, P. J. Slater, Complexity results for closed neighborhood order parameters, Congr. Numer. 112 (1995) 83–96.

[41] T. Tamizh Chelvam, Efficient open domination in Cayley graphs, Appl. Math. Lett. 25 (2012) 1560–1564.

[42] D. T. Taylor, Perfect r-codes in lexicographic products of graphs, Ars Combin. 93 (2009) 215–223.

[43] B. Xue, L. Zuo, G. Wang, G. Li, Shortest paths in Sierpiński graphs, Discrete Appl. Math. 162 (2014) 314–321.

[44] J. Žerovnik, Perfect codes in direct products of cycles—a complete characterization, Adv. in Appl. Math. 41 (2008) 197–205.